1.实验名称及目的

1.1 实验名称

RflySim平台仿真livox mid360激光雷达例程

1.2 实验目的

- 基于仿真环境完整跑通 Livox Mid360 激光雷达的数据链路：从传感器仿真到 ROS 可视化。
- 使用 main.py 自动发现仿真端 IP、配置 UDP 通信协议、初始化传感器并发布 ROS 话题。
- 通过 PX4MavCtrlV4 实现 MAVLink Offboard 起飞与位置控制，验证与仿真平台的联调能力。
- 使用 rflysim.rviz 在 RViz 中查看点云，确认 Config.json 中的传感器配置生效。

1.3 关键知识点（解析本例 *.py 程序）

本例核心逻辑集中在 main.py，涉及“仿真发现与协议设置、传感器创建与 ROS 转发、飞控控制”三部分：

仿真发现与协议设置（ReqCopterSim）

- 通过 ReqCopterSim 获取目标飞机的仿真端 IP：getSimIpID，并打印以便排查网络问题。
- 统一设置 UDP 模式为 2（MAVLink Full）：sendReSimUdpMode，保证与飞控侧协议一致。
- 获取本机 IP（host_ip），用于传感器数据回传。

传感器创建与 ROS 转发（VisionCaptureApi）

- 开启 ROS 话题转发：isEnableRosTrans = True。
- 按 Config.json 的配置创建传感器：jsonLoad(1)。其中典型字段含义：
  - TypeID=23：表示 Livox Mid360 传感器类型。

    - TargetCopter、TargetMountType：指定挂载对象与方式。
    - SensorPosXYZ、SensorAngEular：传感器外参（位置与欧拉角）。
    - DataWidth、DataHeight、DataCheckFreq：数据尺寸与检查频率参数。
    - SendProtocol、otherParams：通信与其他扩展参数（如端口等）。
- 指定数据回传 IP：RemotSendIP = host_ip。
- 与仿真器建立取图请求并校验：sendReqToUE4；随后启动数据采集：startImgCap。

飞控控制（PX4MavCtrlV4）

- 初始化与协议对齐：InitMavLoop(udp_mode)。
- 切换 Offboard、解锁电机：initOffboard、SendMavArm(True)。
- 发送位置控制指令：SendPosNED(0, 0, -1, 0)。采用 NED 坐标系，Z 为 Down，Z=-1 表示上升至约 1 m 位置，实现自动起飞验证。

运行链路总览（结合脚本）

- Windows 侧： SITLPosStr.bat 启动仿真平台与飞控模拟（含 QGroundControl 与 RflySim3D）。
- Ubuntu 侧： run.sh 依次启动 roscore、加载 rflysim.rviz、执行 main.py ，以在 RViz 中查看点云效果。

2.实验效果

- RflySim3D 中创建目标飞行器，QGroundControl 自动连接；飞行器解锁并上升至约 1 m 悬停。
- RViz 依据 rflysim.rviz 的配置展示 Livox Mid360 点云，点云随飞行器姿态和位置实时变化。
- 终端输出可见仿真端 IP 与初始化日志，无异常报错。
- 参考上文示意图，如需对比，可使用相同视角观察点云稳定性与覆盖范围。

验收标准（建议）：

- 点云持续刷新（帧率>0），无明显丢帧或卡死。
- 起飞与位置控制指令生效，姿态稳定，无 Offboard 失败报错。
- 传感器位姿修改后（编辑 Config.json），RViz 中点云有相应变化。

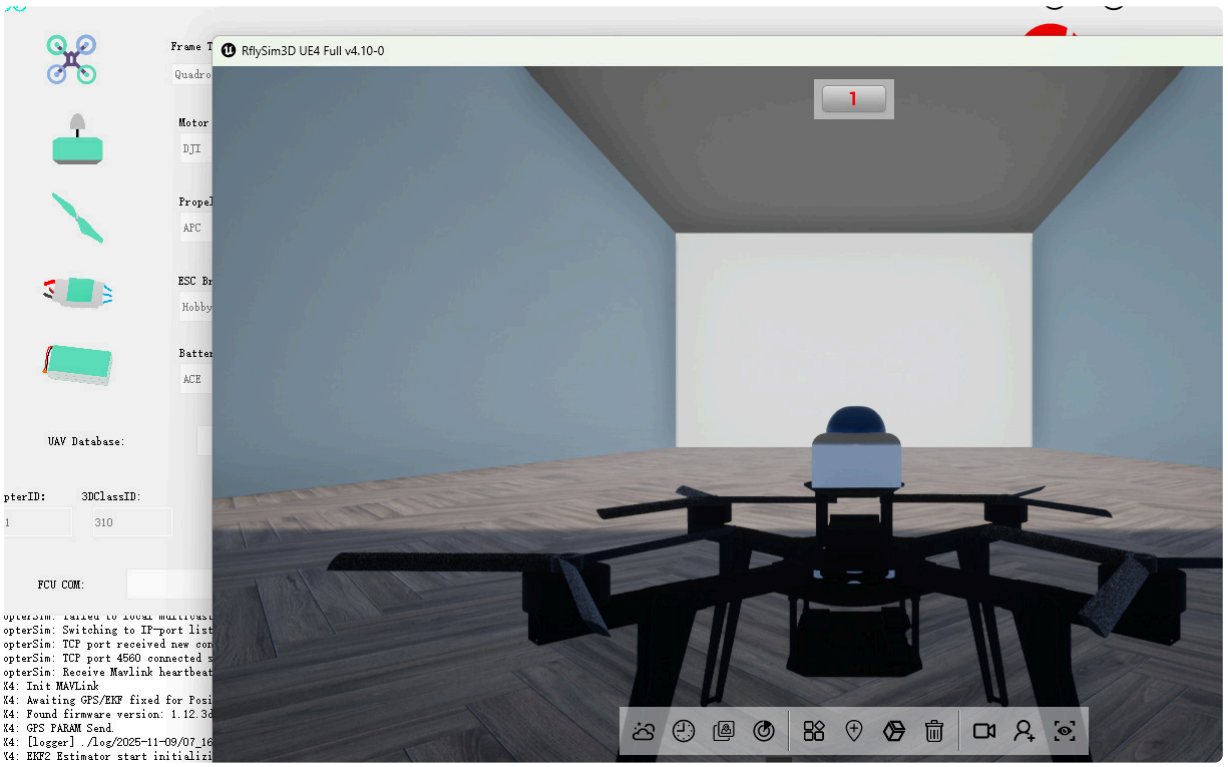
3.文件目录

例程目录： [安装目录]\RflySimAPIs\8.RflySimVision\0.ApiExps\10.Mid360Demo

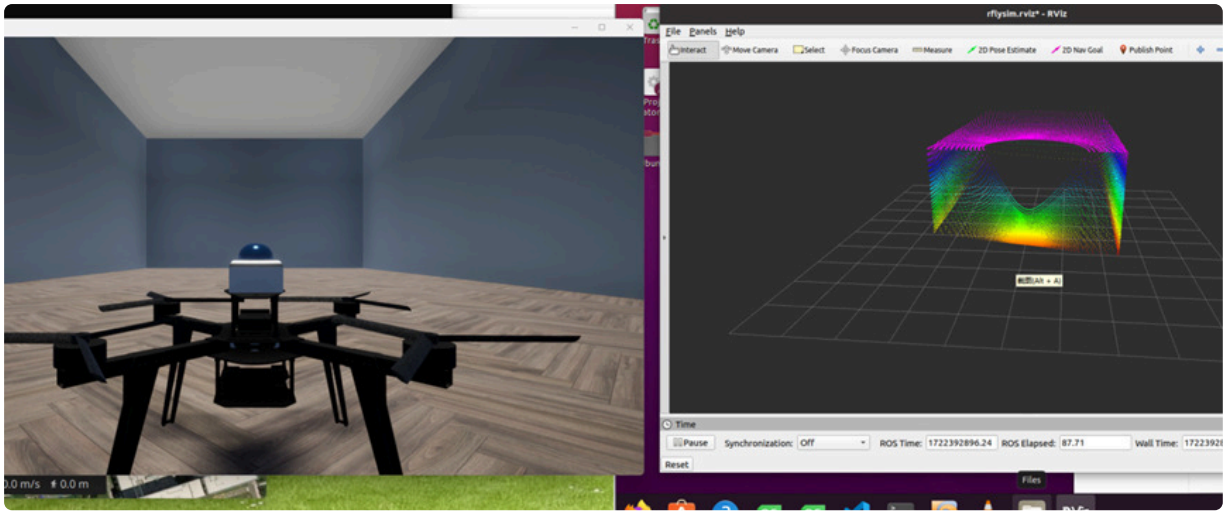
4.运行环境

5.实验步骤

1. 在Windows 端运行SITLPosStr.bat 脚本



2. 在Ubuntu端运行./run.sh 脚本，效果如图：



> 注意事项：Ubuntu 与RflySim 平台运行的电脑在同一个局域网内，彼此能ping通

6.参考资料

- Livox Mid-360 用户手册与 SDK 文档： https://www.livoxtech.com/
- PX4 Offboard 模式指南（官方文档）： https://docs.px4.io/
- MAVLink 协议文档（官方）： https://mavlink.io/
- ROS 可视化工具 RViz 使用文档： http://wiki.ros.org/rviz
- RflySim 平台相关文档与示例： https://rflysim.com/ 或供应商提供的官方文档入口

7. 常见问题（FAQ）

问题：RViz 没有点云显示怎么办?

- 确认 main.py 中已设置 isEnableRosTrans = True。
- 确认已先启动 roscore，并用 rflysim.rviz 打开 RViz 布局。
- 两台设备需在同一网段并可互相 ping 通；防火墙放行传感器使用的 UDP 端口（如 Config.json 中设置的端口）。
- 首次运行无数据可尝试重启 main.py 并观察终端日志中的仿真端 IP 与连接状态。

问题：无法 Offboard 或起飞失败?

- main.py 的 udp_mode=2，应与 Windows 侧的 UDPSIMMODE 保持一致（ SITLPosStr.bat 中为 2）。
- 确认已执行 initOffboard 与 SendMavArm(True)，且仿真侧飞控允许解锁。

- 若仍失败，查看仿真器与地面站状态提示，排查飞控拒绝原因（如模式切换条件不满足）。

问题：获取仿真 IP 失败或连接不稳定？

- 先运行 SITLPosStr.bat，待仿真平台与飞控完全启动后，再在另一台设备执行 run.sh。
- 结合终端打印的仿真端 IP，检查网络配置与连通性；必要时统一固定网段并关闭不必要的 VPN/代理。

问题：Python 依赖缺失导致无法导入模块？

- 安装常见依赖（如 cv2、numpy）；与本例配套的 SDK 模块需正确安装或放置到 Python 搜索路径中。
- 如需使用 MAVLink 相关功能，可按需安装 pymavlink、pyserial。

问题：RViz 布局未生效或需要手动选题？

- 使用 rflysim.rviz 打开默认布局；如需手动选择，请在 RViz 中选择与传感器对应的点云话题（由 VisionCaptureApi 发布）。